Data Structures and Algorithms (DT505G)

First Examination VT2022 (Exam #1)

Teacher:	Federico Pecora (ext. 3319)
Published:	2022-03-25 at 08:15
No. exercises:	6
Total points:	70 (42 required to pass)
No. pages:	10 (excluding this page)

- You **may include images** in your submission (e.g., snapshots of drawings)
- Please make sure that you **number the exercises** in the material you hand in
- **Do not write your name** anywhere
- The teacher will come once to the exam room to clarify any doubts, but if further doubts arise, make reasonable assumptions and **write them down**
- Please use **clear handwriting** in your screenshots
- Answers must be given in **English or Swedish**
- Please use **short and complete sentences**
- All algorithm descriptions must be provided in **pseudocode**

Exercise 1 (10 points)

Given the array $A = \langle 13, 6, 10, 11, 15, 5, 4, 9, 14, 12 \rangle$, show how the Insertion sort and Mergesort algorithms work. Step through the algorithm for a few steps to illustrate how it modifies the input array A. State the worst- and best-case computational complexity of the two algorithms in terms of the size $|A|$ of the input array, and explain why.

Exercise 2 (12 points)

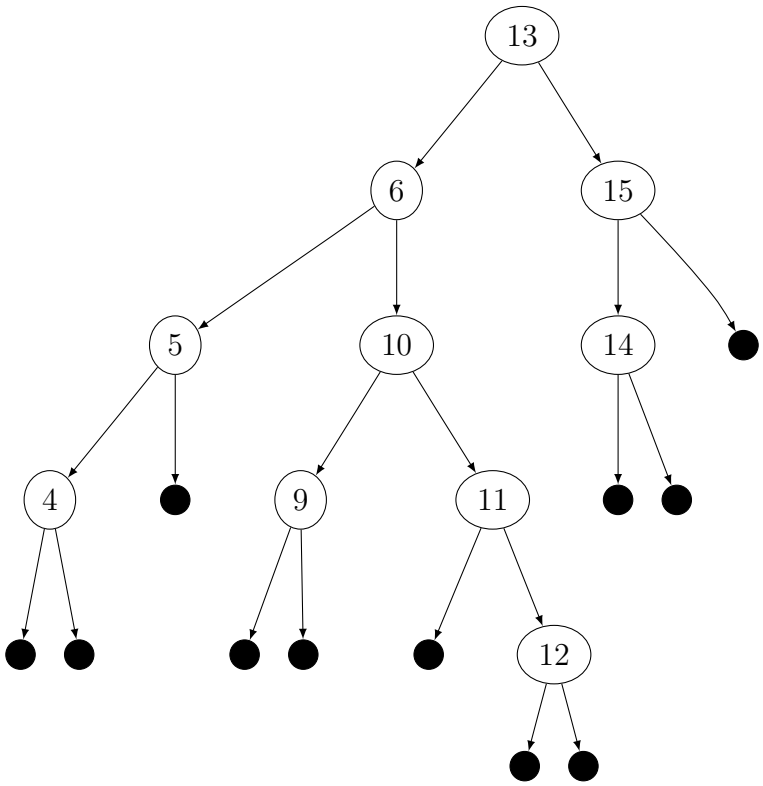
Construct a Binary Search Tree (BST) from the array of integers A in the previous exercise. Do this by inserting each integer into the tree, one by one, starting from the left (element '13'). Draw the resulting tree and state its height. Is the tree balanced?

Remove the third element of A (element '10') from the BST, explaining the operations that this procedure performs. Now, insert the element you removed back into the BST, and draw the resulting tree. Is the resulting BST different from the BST before the element was removed? If so, will this be the case for any element that is removed and then re-inserted?

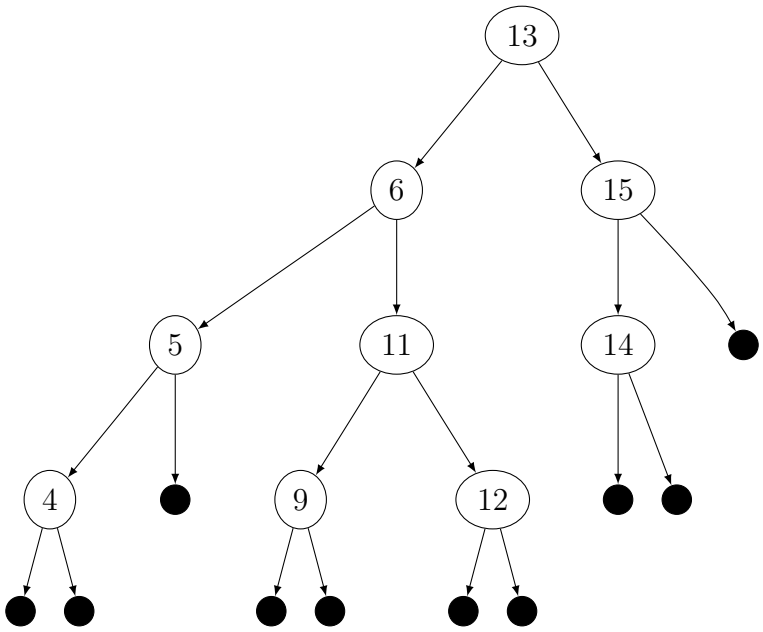
Illustrate the algorithm for finding the predecessor of an element in a BST. Show how this works, step by step, for the fourth element of A (element '11'). In general, how many key comparisons are performed by the algorithm for finding the predecessor of an element in a BST? What is the computational complexity of the algorithm?

Construct a new BST using the sorted array you obtained in Exercise 1, again by inserting the elements of the array into the tree one by one. What is the height of this BST, and is it balanced?

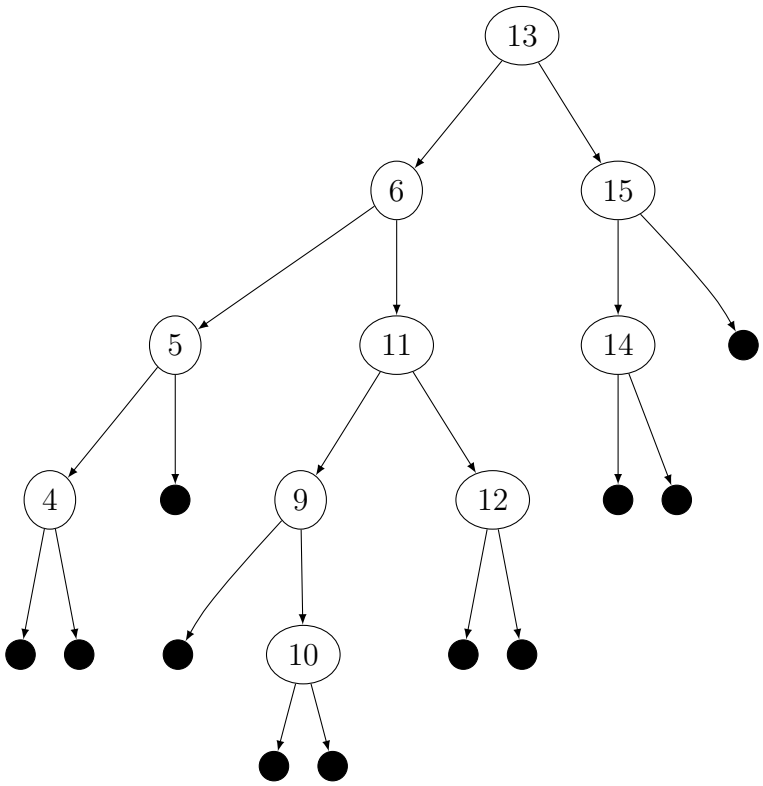
From A:



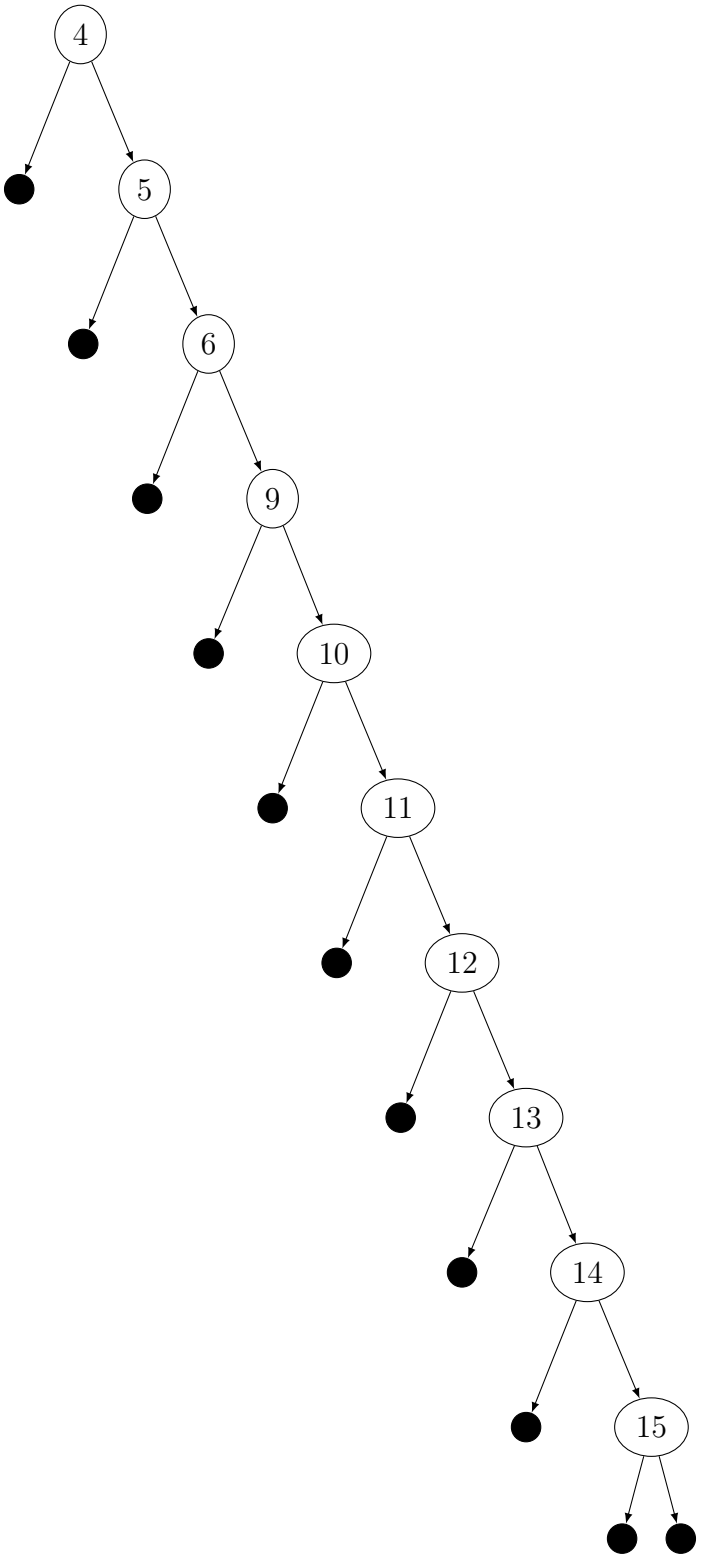
After removing ‘10’:



After re-adding ‘10’:



From sorted array:


4
5
6
9
10
11
12
13
14
15

Exercise 3 (12 points)

The binomial coefficient $\binom{n}{k}$ counts the ways of selecting k elements out of a set of n elements. The binomial coefficient can be defined as

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{otherwise} \end{cases}$$

or, equivalently, as

$$\binom{n}{k} = \begin{cases} 1 & \text{if } k = 0 \\ \left(\binom{n}{k-1}(n-k+1)\right)/k & \text{otherwise} \end{cases}$$

Implement a recursive algorithm for each of the two definitions. Do the two algorithms have the same asymptotic complexity? If not, which one is more efficient? Please explain your answer.

Algorithms:

BINOMIAL_A(n, k)

1 **if** $k = 0$ or $n = k$ **return** 1
2 **return** BINOMIAL_A$(n-1, k-1)$ + BINOMIAL_A$(n-1, k)$

BINOMIAL_B(n, k)

1 **if** $k = 0$ **return** 1
2 **return** (BINOMIAL_B$(n, k-1) * (n-k+1))/k$

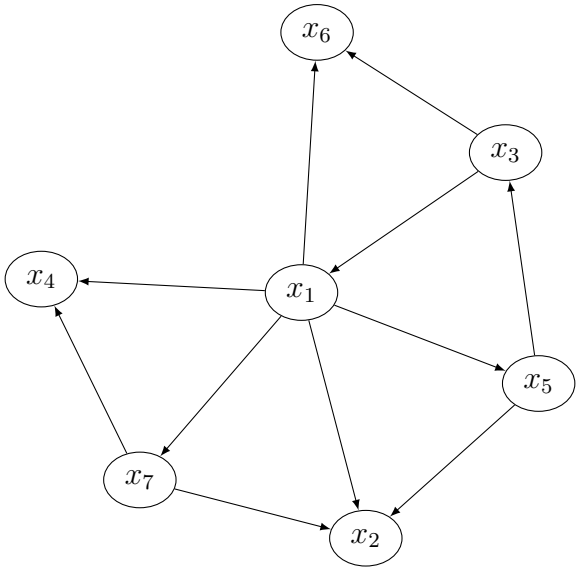
Complexity: The first algorithm computes the binomial by adding ones, so it needs to add $\binom{n}{k}$ ones, hence it has complexity $\Theta(\binom{n}{k})$. The second has complexity $\Theta(k)$ because it performs k recursive calls (decreases k by one at each call). So, the second definition leads to a much more efficient algorithm (note that in the worst case, which is when $n = 2k$, the first one is exponential).

Exercise 4 (12 points)

Given a directed, unweighted graph $G = (V, E)$, describe an algorithm to solve each of the following problems:

- Determine whether the graph is acyclic. DFS (iff no back edges), $\Theta(V + E)$
- Find all the strongly connected components of the graph. DFS + transpose + DFS, $\Theta(V + E)$
- Find the shortest path from node x_1 to every other node. BFS, $O(V + E)$

Show all the steps performed by each algorithm, using as input the graph G given below. Also, state and explain the computational complexity required to solve each problem in terms of the number of nodes $|V|$ and number of edges $|E|$ of the input graph.

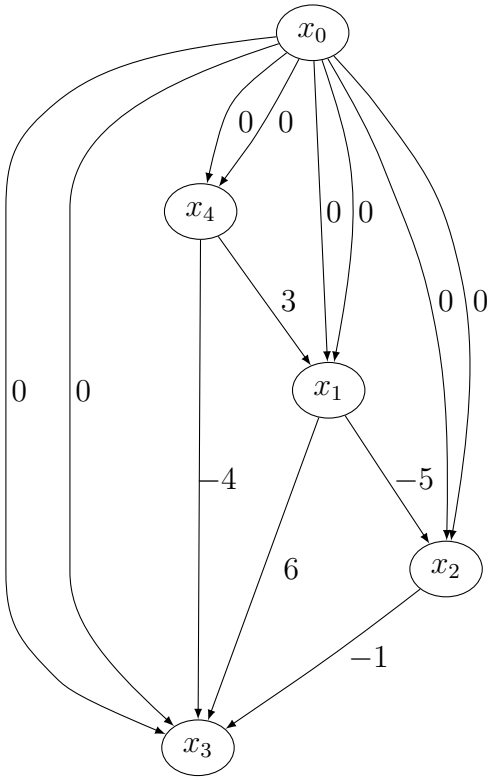


Exercise 5 (12 points)

Illutrate a graph-based algorithm for solving a system of difference constraints with n variables and m constraints. State how the algorithm finds out if the problem is feasible (has a solution), and how it identifies a solution. Show how the algorihtm works on the example below (if the example has a solution, write it down and show how the algorithm finds it; if it does not, show how the algorithm finds out that there is no solution). State the asymptotic complexity of the algorithm.

$$
\begin{aligned}
x_2 - x_1 &\leq -5 \\
x_3 - x_4 &\leq -4 \\
x_3 - x_1 &\leq 6 \\
x_3 - x_2 &\leq -1 \\
x_1 - x_4 &\leq 3
\end{aligned}
$$

We can use Bellman-Ford, whose asymptotic complexity is $\Theta(nm)$. A solution exists iff the distance graph has no negative cycles.



The sytem of difference constraints is feasible and a solution is:

$$
\begin{aligned}
x_1 &= 0 \\
x_2 &= -5 \\
x_3 &= -6 \\
x_4 &= 0
\end{aligned}
$$

Exercise 6 (12 points)

State whether each of the following 6 statements on the asymptotic behavior of the function f is true or false.

1.	$f(n) = 3n\log(n)$,	$g(n) = 4n^3 + 4n^2 + 2n$,	$f(n) \in \Omega(g(n))$	(false)
2.	$f(n) = 9n\log(n)$,	$g(n) = 6n\log(n)$,	$f(n) \in \Theta(g(n))$	(true)
3.	$f(n) = 6\log(n)$,	$g(n) = 5n\log(n)$,	$f(n) \in \Theta(g(n))$	(false)
4.	$f(n) = 9^n$,	$g(n) = 8^n$,	$f(n) \in \Theta(g(n))$	(true)
5.	$f(n) = 9^n$,	$g(n) = 3n^4$,	$f(n) \in \omega(g(n))$	(true)
6.	$f(n) = 2n^2 - n$,	$g(n) = 8n\log(n)$,	$f(n) \in O(g(n))$	(false)